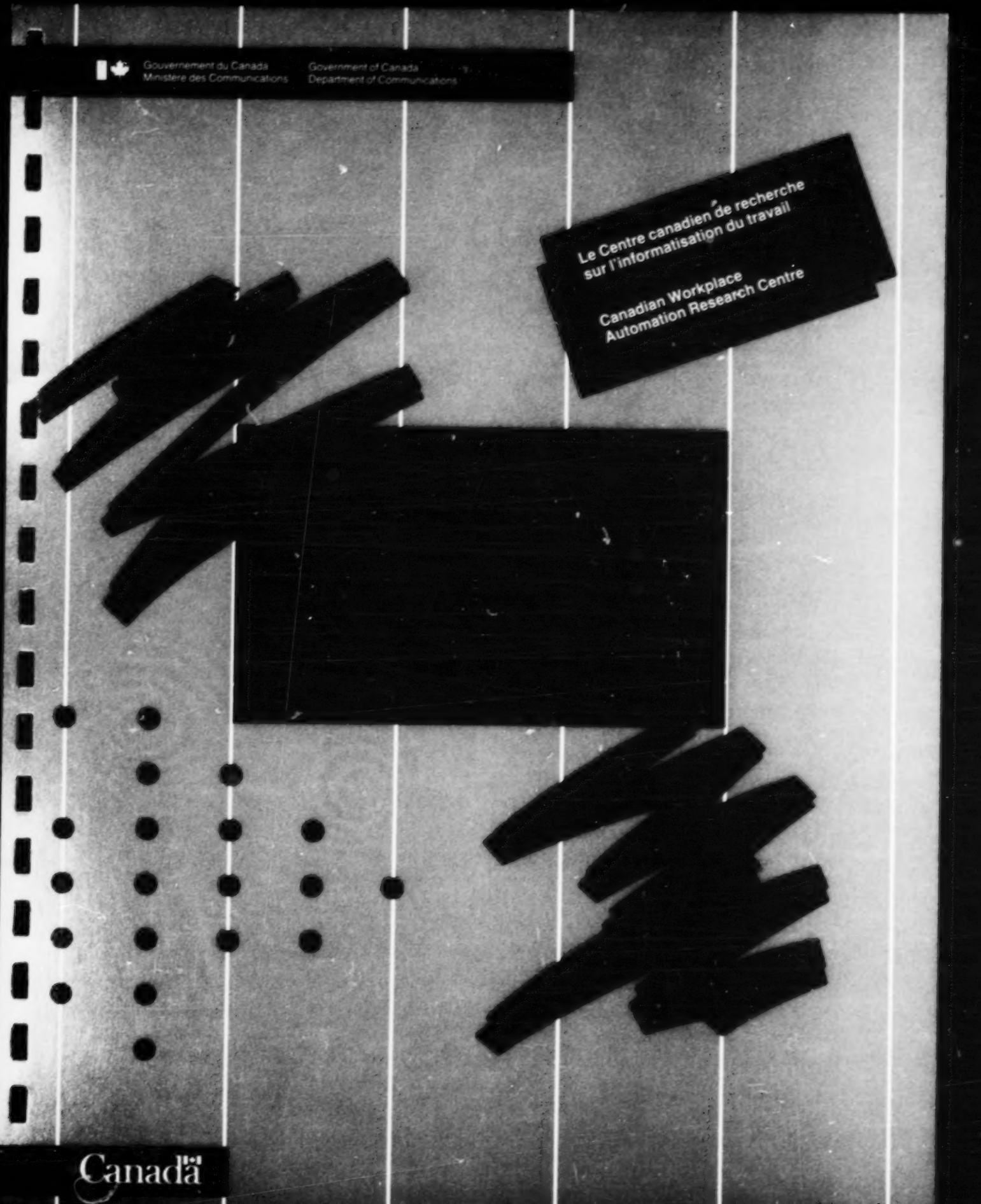

Gouvernement du Canada
Ministère des Communications
Government of Canada
Department of Communications
Le Centre canadien de recherche sur l'informatisation du travail
Canadian Workplace Automation Research Centre
Canada

# CRITTER: un système de traduction pour les rapports sur les marchés agricoles

**Pierre Isabelle**
**Marc Dymetman**
**Elliott Macklovitch**

Centre canadien de recherche sur
l'informatisation du travail
Ministère des Communications

Laval

Octobre 1988

N° de cat.
ISBN

Ce rapport a été présenté à la 12e International Conference on Computational Linguistics, tenue à Budapest, Hongrie, du 22 au 27 août 1988.

Les opinions émises dans ce rapport n'engagent que les auteurs.

* This report is also available in English.

## Abstract

Le système CRITTER en cours de développement servira à traduire des rapports sur les marchés agricoles entre l'anglais et le français. Il repose sur un modèle de transfert et sera réversible. Les textes en langue source et cible sont décrits au moyen a) d'une représentation syntaxique de surface consistant en un arbre annoté de structures de traits, construite avec une grammaire d'extraposition et b) d'une représentation sémantique des structures prédicat/argument, avec vérification des types sémantiques, construite par composition, parallèlement avec la structure syntaxique. La réalisation de CRITTER n'est pas tout à fait terminée, mais les résultats obtenus jusque-là sont prometteurs.


Vedettes matière : Traduction automatique, grammaires logiques.

## 1. Notre approche du problème de traduction

Avec le système de traduction que nous développons, nous visons surtout deux objectifs : a) traduire de l'anglais au français (et du français à l'anglais) un corpus de textes opérationnels d'une sous-langue limitée (Kittredge & Lehrberger, 1982); b) offrir un banc d'essai pour un modèle de traduction reposant sur des bases théoriques motivées : autant que possible, les conceptions retenues devraient tenir compte des dernières réalisations en théorie linguistique et sémantique.

Nous avons choisi pour corpus une série de rapports hebdomadaires publiés par le ministère canadien de l'Agriculture sur les marchés de la viande et du bétail dans les différentes provinces du Canada. Voici un court extrait de ces rapports dans les deux langues :

*Imports of slaughter cattle from the United States last week dropped 62% compared to the previous week, totalling 334 steers and 50 heifers.*

*La semaine dernière, les importations de bovins d'abattage ont chuté de 62% en regard de la semaine précédente, totalisant 334 bouvillons et 50 taures.*

En termes de structure générale, notre modèle de traduction peut être envisagé comme la composition de trois relations abstraites :

(i) la relation analyse/synthèse source :

anasynt_s(T_S, SurfSyn_S, Sem_S)

qui définit un ensemble de triplets bien formés où T_S est un texte en langue source, SurfSyn_S et Sem_S sont respectivement une structure de surface et une structure sémantique pour ce texte, toutes deux dépendant de la langue source;

(ii) la relation analyse/synthèse cible :

anasynt_t(T_T, SurfSyn_T, Sem_T)

qui est l'analogue de anasynt_s pour la langue cible et

(iii) la relation de transfert :

tr(Sem_S, Sem_T)

qui définit un ensemble de couples où Sem_S et Sem_T sont respectivement des structures sémantiques source et cible qui sont considérées équivalentes au niveau de la traduction.

Sur le plan formel et computationnel, les relations anasynt_s et anasynt_t sont décrites par des grammaires d'extraposition (Pereira 1981), et la relation *tr* est définie à l'aide de clauses définies. Il est à noter que chacune des relations anasynt est définie de manière à être réversible (voir la condition de réversibilité dans Landsbergen 1987). Du point de vue pratique, cela veut dire qu'un seul système suffit pour la traduction de l'anglais au français et du français à l'anglais. Du point de vue théorique, la réversibilité est un très fort critère d'adéquation linguistique pour les grammaires. Les analyseurs actuels utilisent généralement des grammaires surgénératrices (parfois très fortement), le "grammairien" tenant pour acquis que le texte à traduire a un degré de

grammaticalité élevé. Au contraire, les générateurs actuels tendent à sous-générer : le grammairien effectue des choix arbitraires dans le système de paraphrases de la langue. Une grammaire réversible est nécessairement plus proche de l'adéquation observationnelle.

## 2. Représentations

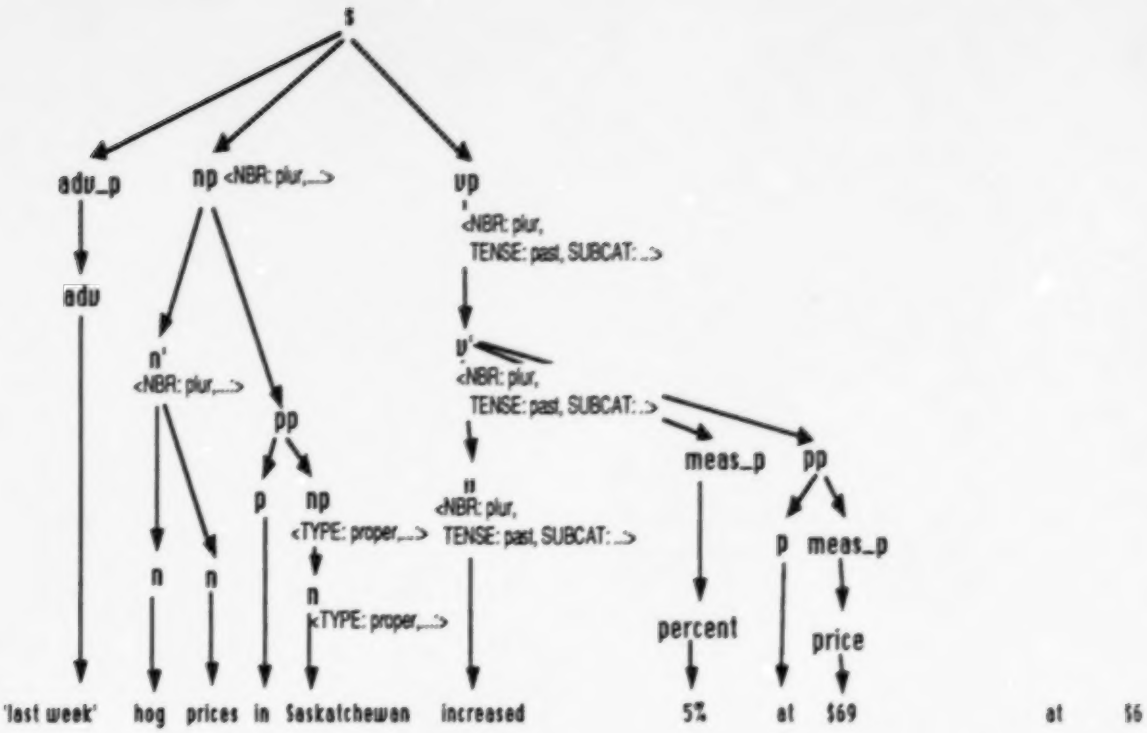
Le système CRITTER affecte à chaque unité textuelle une représentation qui décrit à la fois sa forme (graphologique, morphologique, syntaxique) et son contenu sémantique.

### 2.1 Représentations syntaxiques

La représentation syntaxique associée à un objet textuel est une structure de surface en arbre assez standard, qui peut comprendre des traces dans les cas de dépendances non bornées. Comme nous adoptons une vue monocouche de la syntaxe, nous n'avons prévu aucun autre niveau de représentation syntaxique. Le schème de représentation est une variante de la structure de traits (Sag et al., 1986). Chaque noeud de l'arbre de surface est représenté comme une structure de traits avec, entre autres, des attributs de catégorie (cat) et de filiation (fils).

Dans la notation arborescente classique, nos grammaires courantes donneraient à la phrase 2.a la représentation 2.b:

(2.a) Last week, hog prices in Saskatchewan increased 5% at $69.00.




Cette structure est plus ou moins conforme aux théories syntaxiques courantes. Par exemple, elle reflète une convention X-bar à trois niveaux. A l'occasion, des traits idiosyncratiques sont adoptés pour prendre en compte des particularités de la sous-langue traitée. C'est le cas des compléments *meas_p* et *pp* sous *v'*, qui ne correspondent pas à la sous-catégorisation généralement retenue pour le verbe 'increase'.

## 2.2 Représentations sémantiques

Du point de vue formel, nos représentations sémantiques sont des arbres - ou plus exactement des graphes acycliques orientés, le partage des structures étant permis dans le cas de coréférence - dans lesquels les noeuds sont étiquetés avec des unités sémantiques qui correspondent

souvent, mais non nécessairement, aux lexèmes de la langue représentée. Nous introduisons des unités sémantiques abstraites pour tenir compte des trous lexicaux, des notions sémantiques morphologiquement marquées, etc. Les arcs sont étiquetés soit avec des numéros d'argument (1), (2), (3) ..., soit avec des numéros d'argument inverses (inv-1), (inv-2), ...

On comprendra plus facilement cette notation en examinant la structure sémantique de la figure a), qui est une représentation de la phrase 2.a) : "Last week, hog prices in Saskatchewan increased 5% at $69":

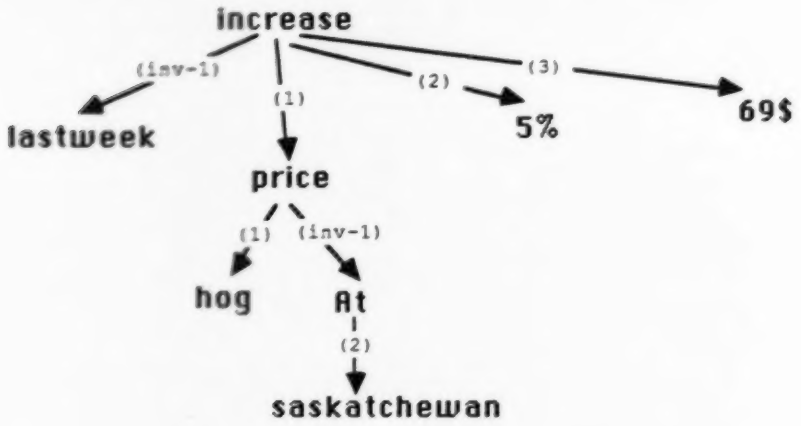



Figure (a)

Dans cette structure :

- 'At', '5%', '69$' sont des unités sémantiques abstraites;

- 'lastweek' est traité comme unité simple, ce qui se justifie par le fait que ce mot joue le rôle d'une expression indiciaire figée dans notre sous-langue (comme 'yesterday' dans la langue normale);

- les étiquettes (1), (2), (3) correspondent à des positions d'argument soit de prédicat (comme 'increase', qui est traité comme un prédicat à trois places) soit de fonction (comme 'price', qui prend une marchandise comme premier argument);

- les étiquettes (inv-1) corresondent à des relations d'arguments inversés, ce qui implique que 'increase' est en premiere position d'argument par rapport à 'last week', et 'price', en première position d'argument par rapport à 'At' ('Saskatchewan' étant le deuxième argument de 'At').

Les étiquettes de type "inv" sont un moyen de notation qui permet de dégager deux niveaux de représentation d'une structure sémantique simple : un premier niveau qui exprime les relations prédicat-argument et un deuxième, qui rappelle la subordination des groupes syntaxiques. Ainsi, 'last week' est un dépendant syntaxique de 'increased' et 'in Saskatchewan', un dépendant syntaxique de 'prices'. Deux raisons nous ont amenés à rappeler la subordination dans la structure sémantique : premièrement, nous voulons que la structure sémantique conserve l'aspect d'un arbre (racine unique, sans cycles). Techniquement, ceci rejoint le fait que le transfert compte essentiellement sur la traversée récursive des structures sémantiques de la racine aux feuilles. Deuxièmement, fait beaucoup plus important, la subordination véhicule de l'information sémantique, mais l'on ne comprend pas bien encore de quelle façon.

Les structures sémantiques doivent se conformer à un critère de bonne formation, qui consiste en la vérification de la compatibilté des types sémantiques entre le prédicat (ou noeud fonctionnel) de la structure et ses arguments. Pour définir la bonne formation sémantique, il faut un *lexique sémantique*, une *hiérarchie des types sémantiques* et des *règles sémantiques*. Le lexique et les règles sémantiques sont brièvement décrits aux sections 3.3 et 4.3 ci-dessous.

## 3. Le lexique

La composante lexicale de CRITTER comprend un dictionnaire d'unités lexicales morphosyntaxiques, des règles qui complètent le dictionnaire morphosyntaxique et un dictionnaire d'unités sémantiques.

### 3.1 Le dictionnaire morphosyntaxique

Le dictionnaire donne une liste d'unités lexicales en forme lemmatisée et leur attribue des propriétés morphologiques et syntaxiques. Il réalise également la projection de ces unités lexicales en unités de niveau sémantique, dont les propriétés sont décrites dans un dictionnaire à part. Dans les propriétés morphologiques sont comprises la classe flexionnelle et l'indication de toutes les idiosyncrasies morphosyntaxiques. Les propriétés syntaxiques regroupent le cadre de sous-catégorisation et une collection de traits syntaxiques. Le *cadre de sous-catégorisation* d'une tête lexicale indique le nombre et le type syntaxique des syntagmes régis par cette tête. Ces cadres renvoient directement à des positions de la structure de surface, étant donné que c'est le seul niveau de représentation syntaxique admis dans notre système. Les verbes, par exemple, peuvent être marqués pour un maximum de trois positions : un sujet et jusqu'à deux compléments.

La projection en unités sémantiques se fait par l'association de chaque entrée lexicale à un schème sémantique. Celui-ci est composé d'une unité sémantique (représentée comme un élément logique à arité fixe), et de l'indication de la relation entre les arguments et les dépendants syntaxiques de l'unité lexicale.

Dans chaque entrée lexicale, ce complexe d'information morphologique, syntaxique et sémantique est spécifié comme une structure de traits. Celle-ci est codée comme un terme Prolog que nous décrivons indirectement, avec des prédicats qui appellent les valeurs d'attributs appropriés. Par exemple, les propriétés syntaxiques et sémantiques du verbe 'promise' pourraient être représentées comme un terme T, décrit comme suit :

(3.a)

```
citation_form(T, promise),
subcat(T, [NP1, NP2, VCOMP]),
cat(NP1, np),
sem_form(NP1, A),
cat(NP2, np),
sem_form(NP2, B),
cat(VCOMP, vp),
vform(VCOMP, infinitive),
sem_form(T, promise'(A, B, C))
```

Les prédicats *citation_form*, *subcat*, *cat* et *sem_form* appellent simplement la valeur d'un attribut du même nom dans le terme T. L'attribut *subcat* a une valeur du type liste, comme dans Pollard & Sag (1988). Les règles syntaxiques vont unifier les éléments de cette liste avec les compléments de la tête lexicale, conformément aux restrictions de sous-catégorisation appropriées (par exemple, la "cat" du deuxième complément de 'promise' est vp).

Ensemble, les attributs *subcat* et *sem_form* assurent une partie essentielle de la projection syntaxe-sémantique. Selon la description donnée pour 'promise', les objets sémantiques associés au sujet, au complément d'objet et au complément infinitif deviennent ainsi respectivement les premier, deuxième et troisième arguments du prédicat 'sémantique promise'.

On peut faire appel à toutes les ressources de la logique clausale pour réaliser des relations complexes entre plusieurs structures de traits. Par exemple, le prédicat *control* utilisé et 3.a) ci-dessus est défini de telle sorte que les valeurs de *sem_form* et *agree* du contrôleur

correspondent à ceux de la case du sujet dans la sous-catégorisation du syntagme verbal contrôlé :

(3.b)

```
control(CONTROLLEE, CONTROLLER) :-
 subcat(CONTROLLEE, [SUJ, X, Y]),
 sem_form(SUJ, SF),
 sem_form(CONTROLLER, SF),
 agree(SUJ, AG),
 agree(CCNTROLLER, AG).
```

Le prédicat *control* en 3.a) garantira donc que le sujet de 'promise' et le sujet sous-entendu de son complément infinitif sont la même entité.

3.2 Règles morphologiques et lexicales

Au dictionnaire morphosyntaxique s'ajoutent trois ensembles de règles qui traitent la flexion, la dérivation et les transformations lexicales. Notre description de la flexion du français est basée sur les travaux de Bourbeau & Pinard (1986) où l'on donne une liste exhaustive des propriétés flexionnelles de plus de 50 000 items lexicaux français. Nous avons développé une description parallèle de la flexion de l'anglais.

Nous employons en ce moment un traitement à règles de la morphologie dérivationnelle, seulement pour les classes les plus productives, comme les adjectifs comparatifs et superlatifs, les adverbes en 'ly', etc. Par ailleurs, nous utilisons beaucoup les transformations lexicales pour traiter les phénomènes comme la passivisation, la montée sujet à sujet et sujet à objet, l'intransitivisation, l'inversion du datif, etc. Etant donné le schéma décrit ci-haut pour la sous-catégorisation lexicale, la plupart

des transformations lexicales peuvent être vues comme simples modifications de l'attribut de sous-catégorisation de l'entrée lexicale.

Par exemple, étant donné la spécification lexicale 3.c) du verbe de montée d'objet 'believe', la règle 3.d) a pour effet de générer les deux entrées "virtuelles" de dictionnaire 3.e) et 3.f).

```
(3.c)
dict(T) :-
 citation_form(T, believe),
 suj_to_obj_raising_verb(T,A,B),
 sem_form(T, believe(A,B)).


(3.d)
subj_to_obj_raising_verb(T,A,B) :-          % standard form
 subcat(T, [NP, S, [ ]]),
 cat(NP, np),
 sem_form(NP, A),
 cat(S, sbar),
 complementizer(S, that),
 sem_form(S, B).
subj_to_obj_raising_verb(T,A,B) :-          % raising form
 subcat(T, [NP1, NP2, VCOMP]),
 cat(NP1, np),
 sem_form(NP1, A),
 cat(NP2, np),
 sem_form(VCOMP, B),
 cat(VCOMP, vp),
 form(VCOMP, infinitive),
 control(VCOMP, NP2).

(3.e)
citation_form(T, believe),
subcat(T, [NP, S, [ ]]),
cat(NP, np),
sem_form(NP, A),
cat(S, sbar),
complementizer(S, that),
sem_form(S, B),
sem_form(T, believe(A,B)).
```

"Tom believes that Bill is dishonest."

(3.f)

```
citation_form(T, believe),
subcat(T, [NP1, NP2, VCOMP]),
cat(NP1, np),
sem_form(NP1, A),
cat(NP2, np),
sem_form(VCOMP, B),
cat(VCOMP, vp),
form(VCOMP, infinitive),
control(VCOMP, NP2),
sem_form(T, believe(A,B)).
```

"Tom believes Bill to be dishonest."

### 3.3 Le lexique sémantique

Le lexique sémantique définit un ensemble d'unités sémantiques pour chaque langue (réalisées chacune directement par un lexème ou de façon plus abstraite), décrit une *hiérarchie de types sémantiques* (un ordre partiel de types) (Sowa, 1983), et associe à chaque unité sémantique SU un type sémantique initial, avec comme conséquence que SU appartient implicitement à tous les types de niveaux supérieurs de la hiérarchie. Le lexique sémantique définit ainsi un ensemble de schémas de validation prédicat/argument dont les structures prédicats/arguments valides doivent être des instances. Voici un exemple de ces shémas :

MOVEMENT( MEASURE-FUNCTION, INCREMENT, MEASURE)

MOVEMENT, MEASURE-FUNCTION, INCREMENT et MEASURE étant des types sémantiques.

L'usage de ce lexique pour vérifier l'acceptabilité des structures sémantiques est brièvement expliqué en 4.3.

## 4. Les grammaires

### 4.1 Règles syntaxiques

Les grammaires de CRITTER associent aux unités textuelles une structure de traits qui décrit à la fois leur forme syntaxique et leur contenu sémantique. Considérons, par exemple, la règle 4.a) :

(4.a)

```
vbar(VBAR) -->
 vb(VB),
 complement(CO1),
 complement(CO2),
 {cat(VBAR, vbar),
 subcat(VB, [SUJ, CO1, CO2]),
 head_of(VBAR, VB)}.
```

Le constituant *vbar* se réécrit comme un verbe suivi éventuellement de deux compléments.

Grosso modo, un complément peut se réécrire en un grand nombre de syntagmes différents.

(4.b)

```
complement([]) -> [].
complement(NP) -> np(NP).
complement(PP) -> pp(PP).
complement(VCOMP) -> vp(VCOMP).
    .....
```

La plupart des règles syntaxiques que nous utilisons sont, comme la règle 4.a), basées sur le modèle hors-contexte des grammaires à clauses définies avec les mêmes mécanismes d'augmentation : les non-terminaux ont des arguments et des buts PROLOG additionnels (entre accolades) peuvent être inclus. Les non-terminaux de nos règles n'ont qu'un seul argument, dont le contenu est une structure de traits, et les buts PROLOG sont utilisés pour énoncer des contraintes réciproques entre ces structures de traits.

Dans l'exemple 4.a), les deux compléments du verbe sont unifiés avec le second et le troisième éléments de la liste *subcat* du verbe, conformément aux exigences de sous-catégorisation lexicale. Le prédicat *head_of* est défini de telle sorte que les traits de la tête lexicale sont unifiés avec ceux du syntagme verbal dominant. Etant donné que *sem_form* est un trait de tête, la valeur lexicale de *sem_form* du verbe est attribuée au syntagme verbal. Dans le processus, les arguments du prédicat sémantique associé au verbe sont instanciés aux objets sémantiques associés aux compléments du verbe.

Pour traiter certains phénomènes syntaxiques plus complexes, comme les dépendances non bornées, nous nous servons des mécanismes spéciaux intégrés dans le formalisme des grammaires d'extraposition.

## 4.2 Traitement syntaxique

Si le format de nos règles de grammaire se rapproche beaucoup de celui des grammaires à clauses définies (DCG) et des grammaires d'extraposition (XG), il présente cependant des différences importantes.

À cause de leur relation directe avec la logique clausale, les DCG et les XG ont deux interprétations distinctes : une interprétation déclarative où l'on peut les envisager comme définissant une relation entre chaînes de caractères et descriptions de structures et une interprétation procédurale, où l'on peut les considérer indifféremment comme des analyseurs ou comme des synthétiseurs.

Toutefois, avec les compilateurs standard de ces formalismes, l'interprétation procédurale d'un ensemble de règles donné peut rarement être utilisée et pour la tâche d'analyse et pour la tâche de synthèse. Par exemple, toute DCG contenant des règles récursive à gauche produira une boucle infinie si on l'applique à l'analyse; la même grammaire peut fort bien convenir pour la tâche de synthèse. En outre, pour obtenir des analyseurs et des synthétiseurs raisonnablement efficaces, il faut contrôler l'ordre dans lequel les buts sont appelés dans chaque mode.

Notre solution à ces problèmes est de s'en tenir à l'utilisation des règles de type DCG qui ont une sémantique déclarative bien définie, mais en les enrichissant d'annotations de contrôle qui, tout en n'affectant pas leur sémantique, fournissent au compilateur l'information dont il a besoin pour produire à la fois une version de la règle orientée vers l'analyse et une autre, orientée vers la synthèse. La récursivité à gauche est supprimée de la version analyse et les deux versions présentent un ordre différent d'exécution des buts. Ainsi, nous pouvons effectivement créer des analyseurs et des synthétiseurs efficaces à partir d'une seule et même grammaire.

Pour plus de détails sur cette approche de double compilation, voir Dymetman & Isabelle (1988).

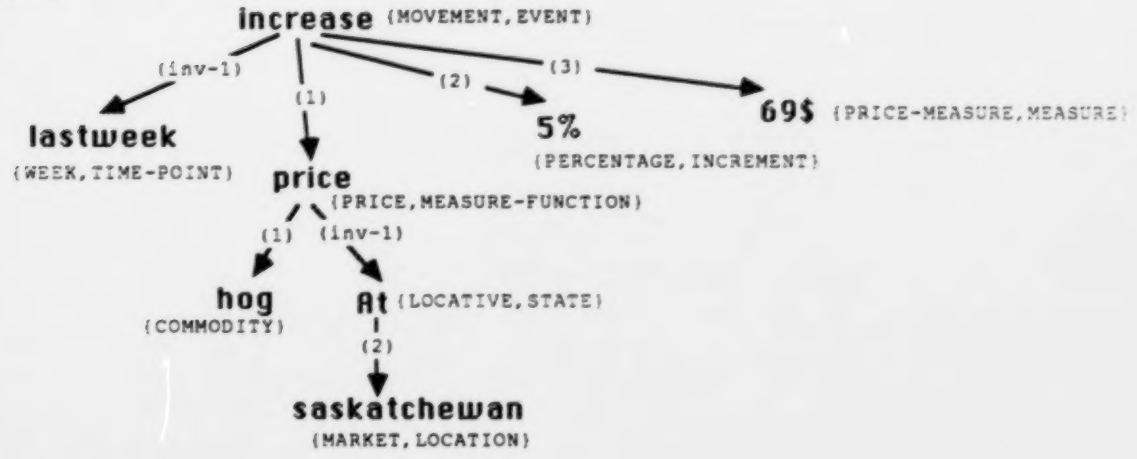
### 4.3 Vérification sémantique

Pour que les structures sémantiques construites par ce processus de composition soient acceptées comme valides, elles doivent passer une *vérification de conformité sémantique*, qui met en jeu des *restrictions sémantiques* et une *hiérarchie des types*.

Cette vérification peut être brièvement décrit ainsi : pour chaque prédicat (ou noeud fonctionnel) *pn* de la structure sémantique, ayant *an1*, *an2* ... comme arguments, l'on doit essayer de trouver un *schéma de validation* (voir 3.3) *PT*(*AT1*, *AT2*, ...) tel que *PT* est un type qui inclut *pn*, *AT*1, un type qui incut *an*1, *AT2*, un type qui inclut *an2*,...

Par exemple, étant donné la structure sémantique de la section 2.2, en l'annotant partiellement avec les types sémantiques, on obtien 4.c).

(4.c)




La vérification de cette structure implique donc la recherche :

- d'un schéma de validation pour 'lastweek', dans ce cas le shéma

TIME-POINT(EVENT)

- d'un schéma de validation pour 'increase', dans ce cas le schéma

MOVEMENT(MEASURE-FUNCTION,INCREMENT,MEASURE)

- d'un schéma de validation pour 'AT', dans ce cas le schéma

AT(PRICE,MARKET).

## 5. Transfert

Comme nous l'avons vu, la composante de transfert met en oeuvre la relation entre deux structures sémantiques, chacune dépendante d'une langue. La décision de limiter l'entrée et la sortie du transfert à de telles structures sémantiques est motivée par un certain nombre de considérations. Tout d'abord, la notion même de traduction implique une reformulation linguistique qui préserve le sens. De telles structures intermédiaires abstraites ont aussi l'avantage pratique de simplifier la composante de transfert. Autrement dit, nous supposons que la composante d'analyse est assez puissante pour neutraliser certaines transformations de la langue source et qu'une composante de synthèse complète peut prendre en charge certains détails de la traduction dans la langue cible comme la réalisation des prépositions régies.

La composante de transfert elle-même est essentiellement lexicale, toutes les connaissances nécessaires étant exprimées dans un *lexique de transfert* dont voici un exemple :

(5.a)

```
(i)   eat <-> manger.

(ii)  miss( 1:  X, 2:  Y )   <-> manquer( 1: Y',  2: X' ).

(iii) walk( inv-1:  across( 2:  X ) )
      <-> traverser( 2: X',  inv-1:  $manner (2: apied) ).
```

L'interprétation de l'entrée (i) est évidente.
L'entrée (ii) exprimes une inversion d'arguments : *john misses mary <=> mary manque à john.*
L'entreé (iii) exprime une correspondance plus complexe : *john walks across the street <=> john traverse la rue à pied.*

Ce lexique est compilé en un ensemble de clauses Prolog. L'algorithme de transfert exécute simultanément une traversée récursive (de la racine aux feuilles) des structures sémantiques source et cible, en utilisant ces clauses pour garder l'équivalence au niveau de la traduction. En pratique, le résultat est le suivant : dans la traduction de l'anglais au français, par exemple, l'algorithme de transfert parcourt la structure sémantique anglaise, la structure sémantique française étant construite parallèlement, par instanciations progressives.

De cette façon le processus de transfert peut réaliser certaines restructurations, mais déclenchées par des entrées lexicales seulement : nous ne prévoyons pas la nécessité d'une composante de transfert structurel indépendante, comme dans ARIANE-78, par exemple (voir Boitet & Nedobejkine, 1981).

## 6. Conclusion

CRITTER est implanté en QUINTUS Prolog sur postes de travail SUN-3. Au moment où cet article est écrit, le prototype se présente comme suit :

- la morphologie de l'anglais et du français est opérationnelle, avec des descriptions exhaustives des systèmes flexionnels; les dictionnaires comprennent approximativement 500 lexèmes dans chaque langue et on s'attend à ce qu'ils dépassent les 1000 lexèmes chacun;

- nous avons déjà des descriptions syntaxiques pour une grande partie des constructions trouvées dans la sous-langue, mais il reste encore beaucoup à faire pur traiter adéquatement les ellipses, les coordinations complexes, etc.; de plus, les grammaires anglaise et française sont utilisées de manière réversible, bien qu'en ce moment elles aient tendance à surgénérer en mode synthèse;

- une version simple de la vérification sémantique des types a été implantée, mais il reste à définir une hiérarchie adéquate des types sémantiques;

- une implantation initiale de la composante de transfert (dictionnaire et programmes) est en cours et les premières traductions devraient sortir dans quelques semaines.

## 7. Remerciements

Les auteurs tiennent à remercier Lucie Cusson, Bruno Godbout, François Perreault et Michel Simard, de l'aide

précieuse qu'ils leur ont apportée dans le travail exposé dans cet article.

## RÉFÉRENCES

Boitet, C. et Nedobejkine, N. "Recent developments in Russian-French machine translation at Grenoble". Dans : Linguistics 19, pp199-271, 1981.

Bourbeau, L. et Pinard, F. Dictionnaire Micro-informatisé du francais. Montreal, Canada, 1986.

Dymetman, M. et Isabelle, P. "Reversible Logic Grammars for Machine translation". Dans Second International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages, Carnegie-Mellon University, 1988.

Gazdar, G., Klein, E., Pullum, G., Sag I. Generalized Phrase Structure Grammar. Oxford: Blackwell, 1985.

Isabelle, P. et Macklovitch, E. "Transfer and MT Modularity". In Proceedings of Coling 86, Bonn, FRG, 1986.

Kittredge, R. et Lehrberger, J. eds.:Sublanguage: studies of language in restricted semantic domains. Berlin: de Gruyter, 1982.

Landsbergen, J. Montague Grammar and Machine Translation. Philips Research M.S. 14.026, Eindhoven, The Netherlands, 1987.

Pereira, F. "Extraposition Grammars". Dans Computational Linguistics 7.4, pp243-256, 1981.

Pollard, C. et Sag, I. Information-Based Syntax and Semantics, vol. 1: Fundamentals, CSLI lecture Notes no 13, CSLI, Stanford University, 1988.

Sag I., Kaplan R., Karttunen L., Kay M., Pollard C., Shieber S., Zaenen A. "Unification and Grammatical Theory", Dans Proceedings of the West Coast Conference on Formal Linguistics, Stanford Linguistics Association, Stanford University, 1986.

Sowa, J. Conceptual Structures: Information Processing in Mind and Machine. Reading, Mass: Addison-Wesley, 1983.